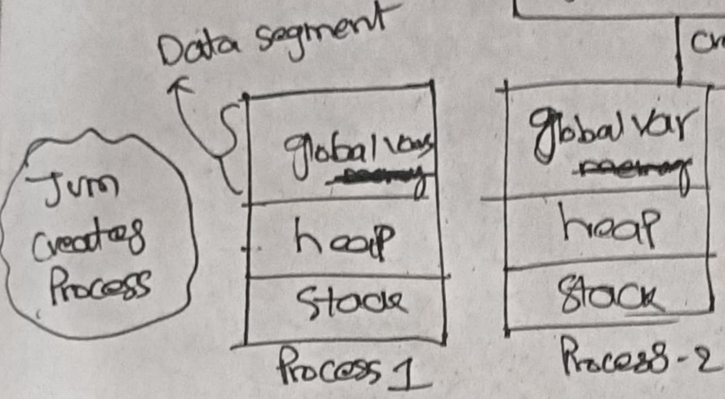
Multithreading

-> Process v/s Thread

Process :- It is an instance of the Program that is being executed.

eg:- Java ClassName

creates JVM Process they maintain Threads



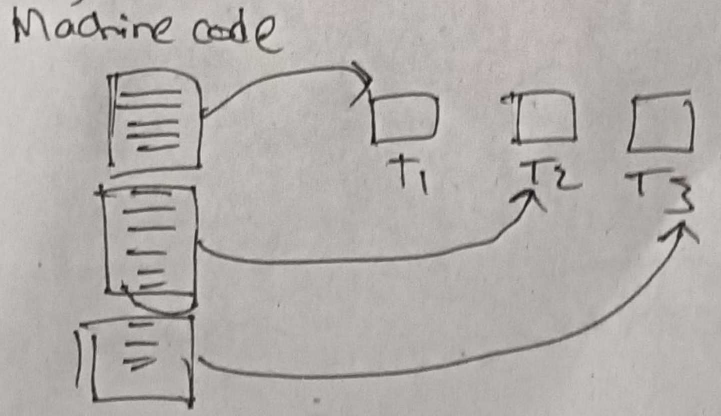
2 Process never Share resources they run Separately

Thread :- It is a light weight Process

(or)

-> Smallest sequence of instructions that are executed by cpu independently

-> Process can have multiple thread.



-> Initialy we have main ~~class~~ Thread from This multiple Threads are Created

eg :- Thread.currentThread().getName();

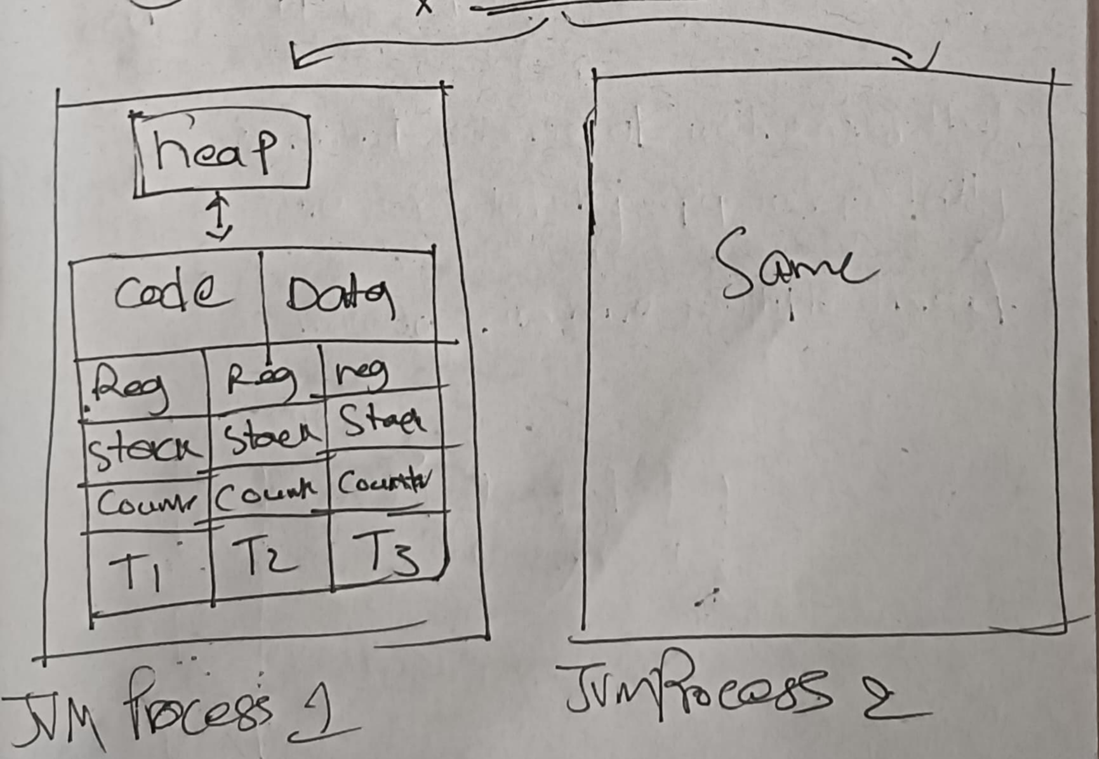
-> Stack
-> heap
-> Code section
-> Data
-> PC
-> Registers

JVM

Java.Myclass :

↓

1. Create new JVM Process



Code	Data	
Reg	Reg	reg
stack	Stack	Stack
Counter	Count	Counter
T1	T2	T3

eg:-

```
java -Xms256m MyClass
java -Xmx2g   Myclass
```

Why Separate registers, Stack, Counter

Code Segment

→ Stores machine code that CPU understands

→ Data Segment (global & Static data)

⊛ → read only

Data Segment

→ Global & Static Vars

→ multiple Threads can access data

→ So Synchronization is needed

⊛ → read/write

Heap

→ "new" Keyword Creates object in heap area

→ Threads Share Same heap memory

→ Synchronization is required

⊛ → read/write

Stack:-

-> Each Thread has own Stack

-> manages method calls, local Variables, return address

Register :-

-> Store intermediate value

-> JIT uses stack to reshuffle the machine code

★ -> Helps in Context Switching

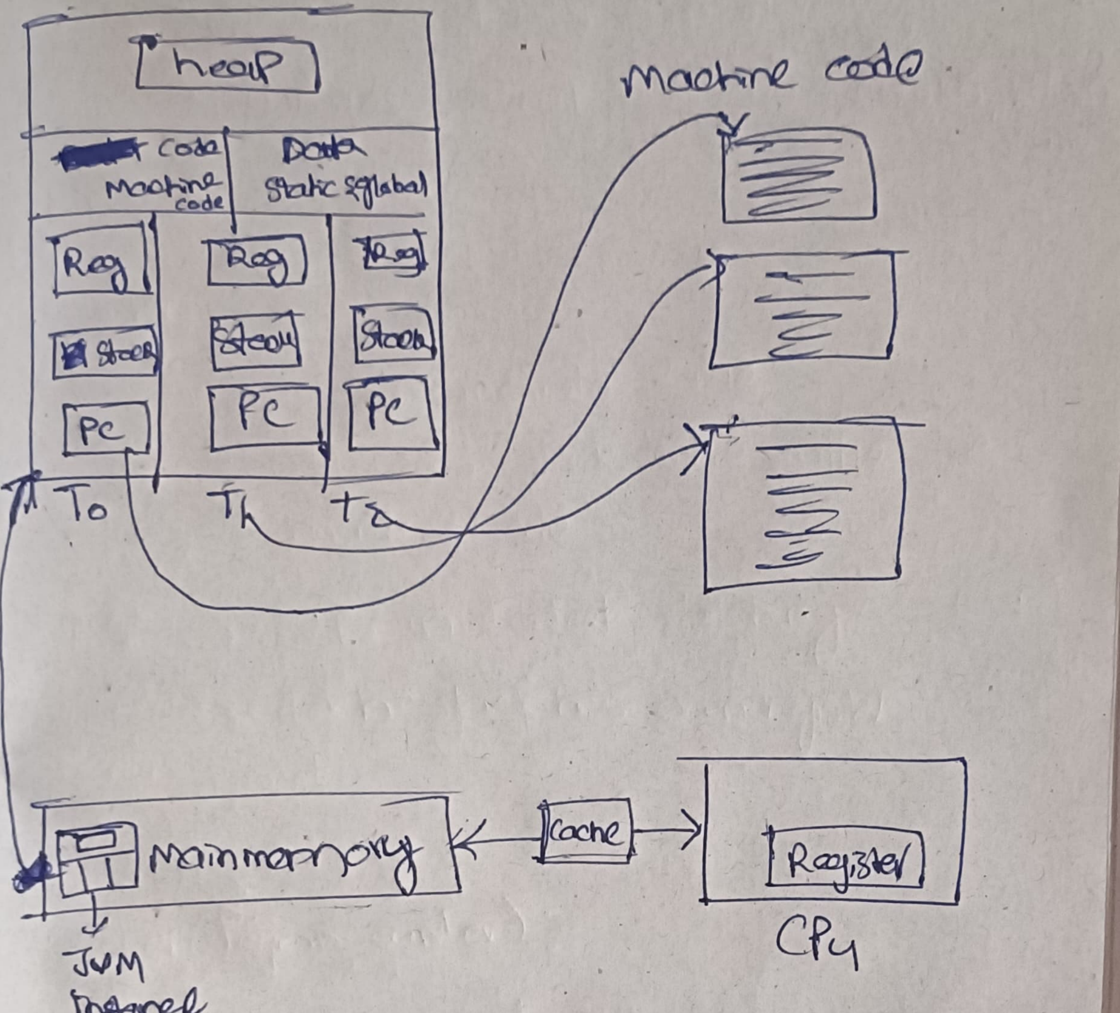
-> Each Stack has own register

Counter (Program Counter) :-

Point to current executing instruction

after execution PC is incremented to next instruction that need to be executed

Java MyClass → JIT Compiler → Machine code



This machine code part which T_0's PC is Pointing to is copied into T_0 register and based on OS Scheduling algs (FCFS, SJF, SRTF, RR) Thread will be allocated execution time and CPU uses its registers for storing intermediate values when its time is up CPU register data copied into T_0's individual register now its turn of T_1. This Process is called Context Switching